Universidad Mariano Gálvez de Guatemala

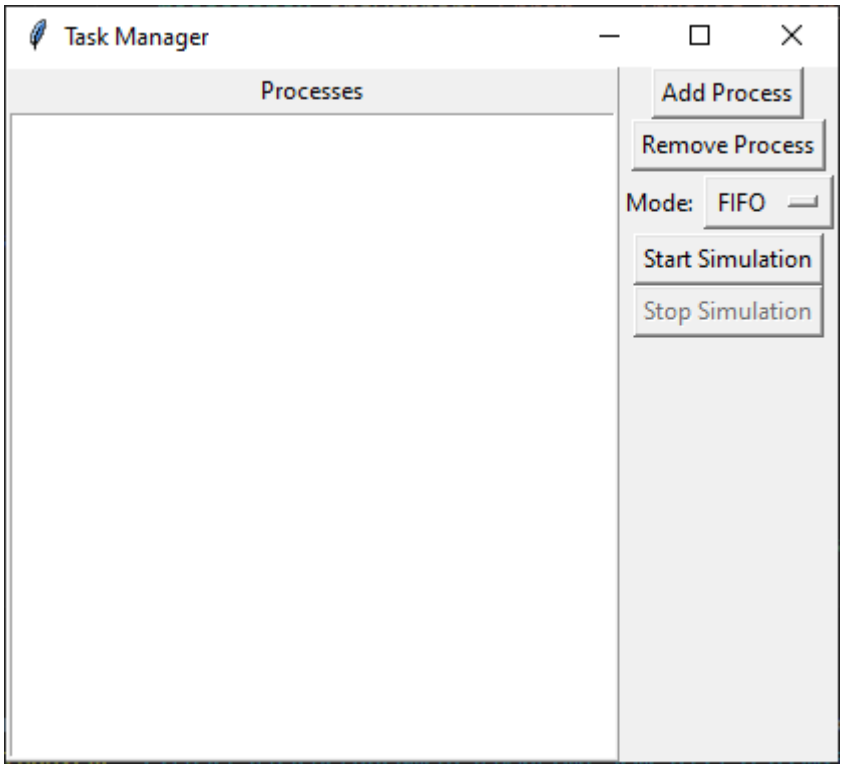
Ingeniería en Sistemas

Sistemas Operativos 2

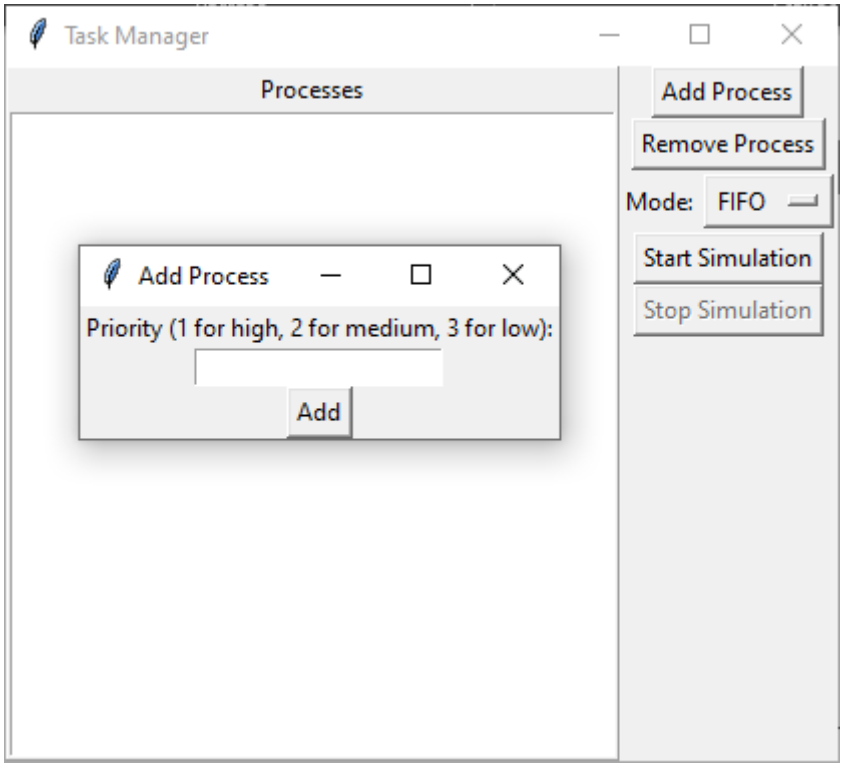
TASK MANAGER

César Antonio Flores Guillén, 1290-21-292

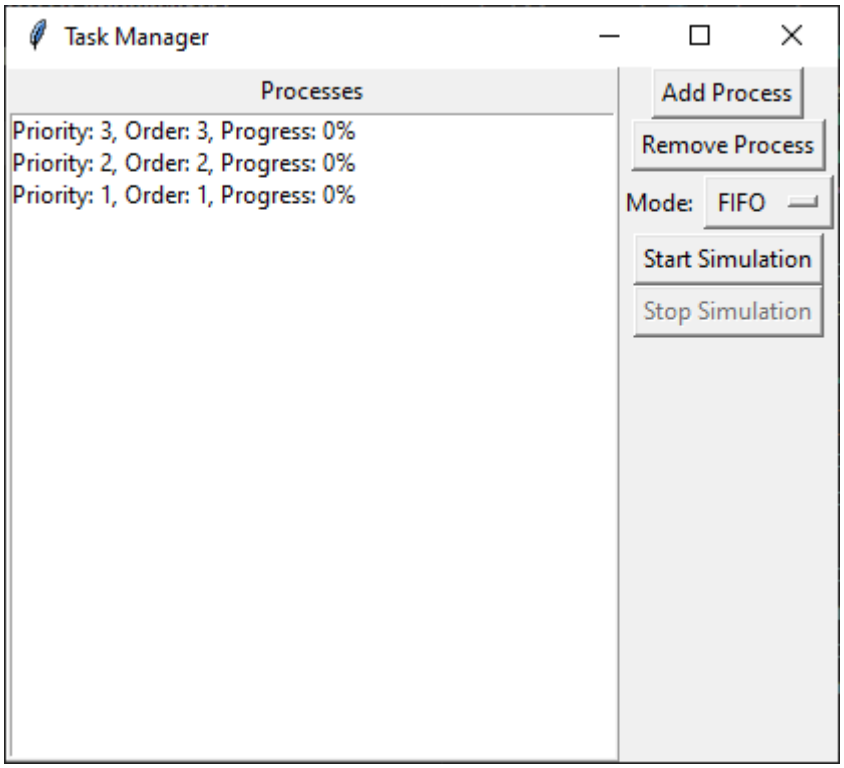
Ejecutas el archivo *task.py*



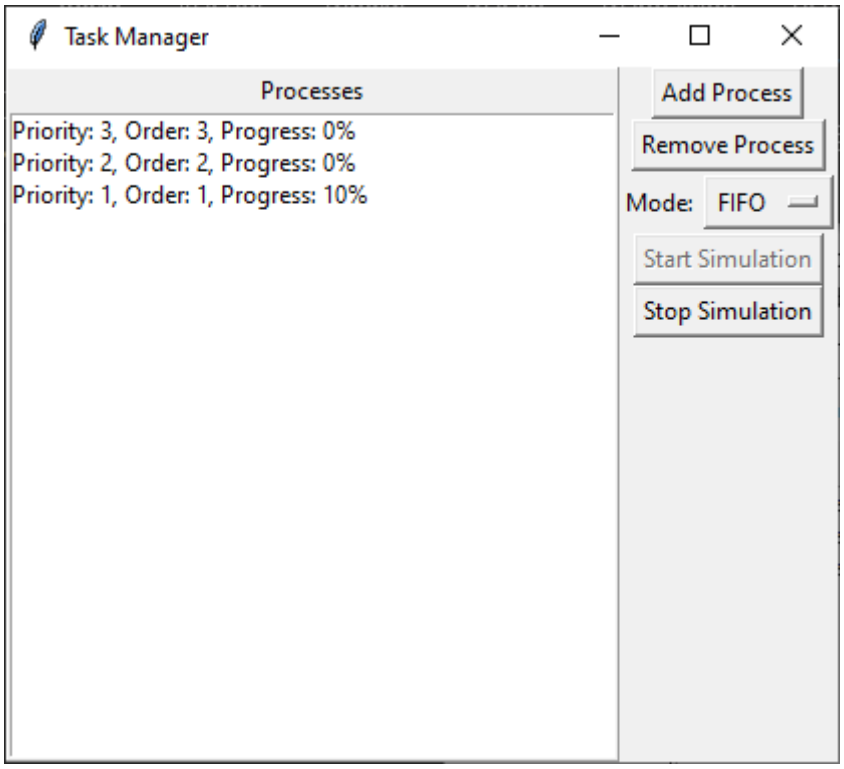
Con el botón *Add Process* se abrirá una nueva ventana



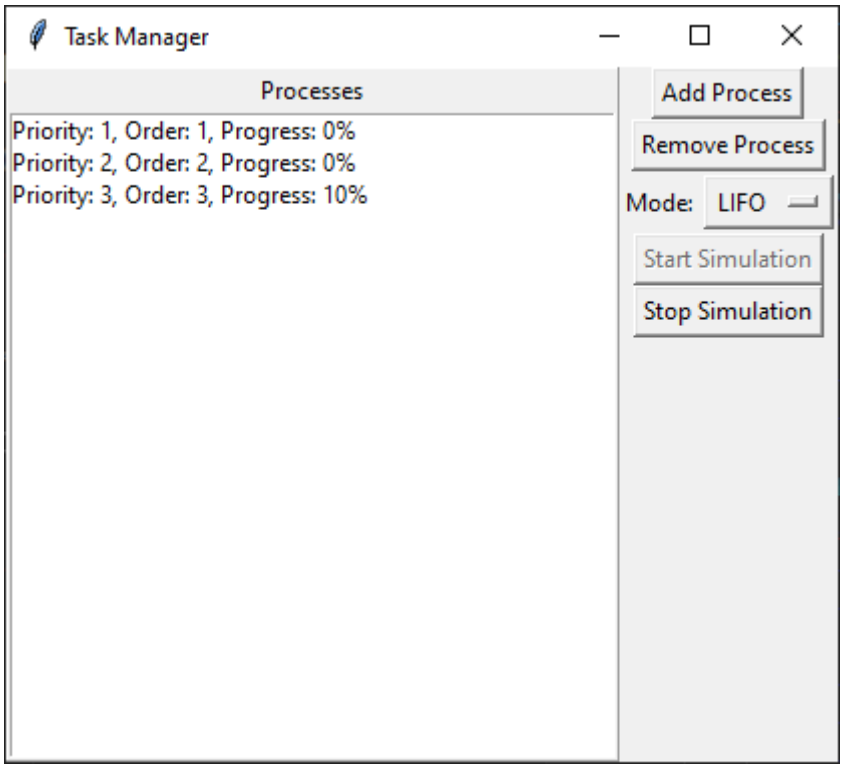
Aquí podremos añadir un nuevo proceso seleccionando su prioridad entre 1 y 3, para el ejemplo añadiremos 1 de cada prioridad. El botón *remove process* elimina los procesos según el modo seleccionado.



Cuando el Modo esta configurado como FIFO, el primer proceso en avanzar será el de *Order: 1*



Cuando está en modo LIFO, el primero en avanzar será el proceso de llegada 3.



Info del código:

1. **Importación de Módulos:**
    - **tkinter as tk**: Importa la biblioteca tkinter y la abrevia como **tk** para su uso en el código.
    - **from tkinter import messagebox**: Importa la función **messagebox** de tkinter para mostrar cuadros de diálogo.
    - **from tkinter import ttk**: Importa el módulo **ttk** de tkinter para usar el widget **Progressbar**. Aunque se importa, no se utiliza en el código actual.
2. **Importación de Módulos de Hilos y Colas:**
    - **import threading**: Importa el módulo threading para manejar hilos y ejecución concurrente.
    - **import queue**: Importa el módulo queue para usar colas, especialmente útil para la gestión de procesos en hilos separados.
    - **import time**: Importa el módulo time para manejar intervalos de tiempo y pausas en la simulación.

3. **Clase AddProcessDialog:**
    - Esta clase define una ventana emergente para agregar procesos.
    - **__init__(self, parent, process_manager)**: El constructor inicializa la ventana con elementos como etiquetas, entrada de texto y un botón para agregar procesos.
    - **add_process(self)**: Método para agregar un proceso a la lista de procesos. Verifica la prioridad ingresada y muestra un mensaje de error si es inválida.
4. **Clase TaskManager:**
    - Esta clase representa la ventana principal de la aplicación de gestión de procesos.
    - **__init__(self, root)**: Constructor que inicializa la ventana principal con botones, etiquetas y marcos para organizar la interfaz.
    - **open_add_process_dialog(self)**: Método para abrir la ventana de diálogo para agregar procesos.
    - **add_process(self, priority)**: Método para agregar un proceso a la lista de procesos, considerando la prioridad y el modo (FIFO o LIFO).
    - **remove_process(self)**: Método para eliminar un proceso de la lista de procesos.
    - **update_list(self)**: Método para actualizar la lista de procesos en la interfaz gráfica.
    - **update_progress_bars(self)**: Método para actualizar las barras de progreso de los procesos en la interfaz gráfica. **(No se logró implementar)**
    - **simulate_processes(self)**: Método que simula el progreso de los procesos, actualizando sus barras de progreso y la lista de procesos. **(No se logró implementar)**
    - **start_simulation(self)**: Método para iniciar la simulación de procesos en un hilo separado.
    - **stop_simulation(self)**: Método para detener la simulación de procesos.
5. **Función main:**
    - Esta función crea la ventana principal (**root**) y la instancia de la clase **TaskManager** para iniciar la aplicación.
    - **if __name__ == "__main__": main()**: Verifica si el script se está ejecutando como el programa principal y luego inicia la aplicación.

Enlace del repositorio: https://github.com/Checha-AFG/TaskManager